09/828,049

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	"20020100030"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/28 14:37
L3	59	(US-20010004755-$ or US-20010044856-$ or US-20010052120-$ or US-20020013937-$ or US-20020029357-$ or US-20020032719-$ or US-20020032822-$ or US-20020040470-$ or US-20020042695-$ or US-20020042807-$ or US-20020049865-$ or US-20020049894-$ or US-20020054536-$ or US-20020104077-$ or US-20020108106-$ or US-20020108107-$ or US-20020112227-$ or US-20020147969-$ or US-20020165848-$ or US-20030009743-$ or US-20030028866-$ or US-20030079210-$ or US-20030097651-$ or US-20020100030-$).did. or (US-5293631-$ or US-5301325-$ or US-5307492-$ or US-5371747-$ or US-5493675-$ or US-5535391-$ or US-5577253-$ or US-5598560-$ or US-5613117-$ or US-5659753-$ or US-5659754-$ or US-5671416-$ or US-5778233-$ or US-5815719-$ or US-5836014-$ or US-5884083-$ or US-6059840-$ or US-6072953-$ or US-6081665-$ or US-6086632-$ or US-6117187-$ or US-6128775-$ or US-6243864-$ or US-6247174-$ or US-6260189-$ or US-6289507-$).did. or (US-6292938-$ or US-6317876-$ or US-6327647-$ or US-6385757-$ or US-6408428-$ or US-6463582-$ or US-6477683-$ or US-6584611-$ or US-6718541-$).did.	US-PGPUB; USPAT	OR	OFF	2005/05/28 16:02

L4	1	L1 and ((variab$2 near2 siz$2) near2 register)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/28 16:03
L5	1	L1 and (((variab$2 or dynamic) near2 siz$2) near2 register)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/28 16:04
L6	14	(((variab$2 or dynamic) near2 siz$2) near2 register$2) and (compiler) and (@ad<"19981010" or @rlad<"19981010" or @prad<"19981010")	US-PGPUB; USPAT	OR	OFF	2005/05/28 17:54
L7	16	((((variab$2 or dynamic) or flexible) near3 siz$2) near2 register$2) and (compiler) and (@ad<"19981010" or @rlad<"19981010" or @prad<"19981010")	US-PGPUB; USPAT	OR	OFF	2005/05/28 18:02
L8	2	((register near2 allocat$3) near2 size) and (compiler) and (@ad<"19981010" or @rlad<"19981010" or @prad<"19981010")	US-PGPUB; USPAT	OR	OFF	2005/05/28 18:05
L9	10	((register near2 allocat$3) near5 size) and (compiler) and (@ad<"19981010" or @rlad<"19981010" or @prad<"19981010")	US-PGPUB; USPAT	OR	OFF	2005/05/28 18:51
L10	1	"5586323".pn.	US-PGPUB; USPAT	OR	OFF	2005/05/28 18:51

09/828,049

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

allocate dynamic size register

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used **allocate dynamic size register**

Found **48,067** of **155,867**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 A fast, memory-efficient register allocation framework for embedded systems
Sathyanarayanan Thammanur, Santosh Pande
November 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 6
Full text available: pdf(1.01 MB) Additional Information: full citation, abstract, references, index terms

In this work, we describe a "just-in-time," <i>usage density-based register allocator</i> geared toward embedded systems with a limited general-purpose register set wherein speed, code size, and memory requirements are of equal concern. The main attraction of the allocator is that it does not make use of the traditional live range and interval analysis nor does it perform advanced optimizations based on range <i>splitting</i> but results in very good code quality. We circumven ...

Keywords: Code generation, compiler optimizations, compilers, dynamic compilation, embedded systems, register allocation

2 Energy-aware systems: Binary translation to improve energy efficiency through post-pass register re-allocation
Kun Zhang, Tao Zhang, Santosh Pande
September 2004 **Proceedings of the fourth ACM international conference on Embedded software**
Full text available: pdf(190.98 KB) Additional Information: full citation, abstract, references, index terms

Energy efficiency is rapidly becoming a first class optimization parameter for modern systems. Caches are critical to the overall performance and thus, modern processors (both high and low-end) tend to deploy a cache with large size and high degree of associativity. Due a large size cache power takes up a significant percentage of total system power. One important way to reduce cache power consumption is to reduce the dynamic activities in the cache by reducing the dynamic load-store counts. In ...

Keywords: cache power consumption, dead registers, register re-allocation, unused registers

3 A simple interprocedural register allocation algorithm and its effectiveness for LISP
Peter A. Steenkiste, John L. Hennessy
January 1989 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 11 Issue 1